Patchwork-Based Multilayer Audio Watermarking

Iynkaran Natgunanathan, *Member, IEEE*, Yong Xiang, *Senior Member, IEEE*, Guang Hua, *Member, IEEE*, Gleb Beliakov, *Senior Member, IEEE*, and John Yearwood

***Abstract*—A multilayer watermarking system is a system that is able to embed watermarks to a host media signal repeatedly in an overlaying manner, without incurring troubles in extracting the watermarks in each layer. In this paper, we present a novel patchwork-based audio watermarking algorithm that can embed and extract watermark bits successfully in such a multilayer framework. In the proposed method, a new watermark embedding algorithm is designed to ensure that the embedded watermarks in a certain layer do not affect the detection of watermarks in other layers. Adding multiple layers of watermark bits inevitably reduces the perceptual quality. However, to minimize the perceptual quality degradation in multilayer watermarking, the audio fragments for watermark embedding are selected from a set of specially arranged discrete cosine transform coefficients of the host audio signal. Watermark embedding is achieved by modifying the mean values of selected sample fragments. With the use of an embedding error buffer, the proposed system can withstand a wide range of common attacks. To maintain the balance between the perceptual quality and robustness, watermark embedding strength is adjusted according to the specific layer used. The proposed multilayer scheme ensures the independence of the processing in different layers. The effectiveness of the proposed system is demonstrated and verified by extensive simulation results.**

***Index Terms*—Audio watermarking, discrete cosine transform, multi-layer, patchwork.**

I. Introduction

REVOLUTIONARY growth in signal processing and information technologies has allowed large scale production, transmission, and distribution of multimedia data through the cyberspace. This inevitably increases the demand for enhanced copyright protection, broadcast monitoring and multimedia property management. Nowadays in a typical commercial distribution network, a multimedia object passes through various layers such as production, regional and country distribution layers (e.g., a multimedia object may be given to a country distributer by the production company via a regional distributor). When a pirated multimedia object is found, it is necessary to identify the source from which the pirate had accessed the multimedia content and trace the distribution path of the multimedia object. In addition to copyright protection related application, multimedia object tracing is necessary in other applications such as broadcast monitoring and multimedia property management [1], [2]. Fig. 1 illustrates an example of a simple distribution network with possible pirated copy leakage paths. Even in this simplified example, it will be difficult to identify the source of leakage when a pirated copy is found.

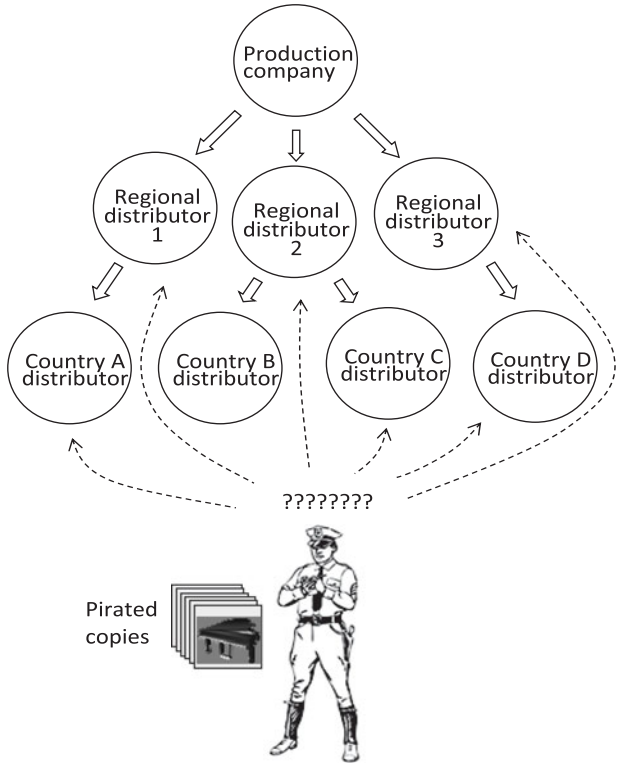

Fig. 1. Example of a commercial multimedia distribution network with possible pirated copy leakage paths.

In order to achieve multimedia object identification and tracing in a distribution network, information associated with a multimedia object such as digital signature, logo and distributor ID should be added to the multimedia object at different layers of the distribution network. For example, while production company may include digital signature and logo, the regional and country distributors may include their respective distributor IDs. To make it feasible for real world applications, this information adding mechanism should satisfy the following requirements.

1) The added information should not significantly degrade the perceptual quality.
2) Added information to a particular layer should not interfere with information added in the previous layers.

Manuscript received March 15, 2017; revised July 8, 2017 and August 27, 2017; accepted August 29, 2017. Date of publication September 4, 2017; date of current version September 20, 2017. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Federico Fontana. *(Corresponding author: Yong Xiang.)*

I. Natgunanathan, Y. Xiang, G. Beliakov, and J. Yearwood are with the School of Information Technology, Deakin University, Melbourne Burwood Campus, Burwood, VIC 3125, Australia (e-mail: iynkaran@deakin.edu.au; yxiang@deakin.edu.au; gleb.beliakov@deakin.edu.au; john.yearwood@deakin.edu.au).

G. Hua is with the School of Electronic Information, Wuhan University, Wuhan 430072, China (e-mail: ghua@ieee.org).

Digital Object Identifier 10.1109/TASLP.2017.2749001

3) It should be possible to successfully extract the information added in a particular layer from a multimedia object which contains several layers of added information.
4) The added information should not be removed by a series of common attacks such as noise addition, compression, re-sampling and filtering, etc.

Although object identification and tracing is necessary in several multimedia data formats such as audio, image and video, in this paper we limit our attention to audio data. Digital watermarking is a smart technology in which the information associated with the multimedia data can be embedded into a multimedia object without significantly affecting its perceptual quality. In the past decade, there are several audio watermarking algorithms developed based on techniques such as amplitude modification [3], patchwork [4]–[7], echo-hiding [8]–[15], spread spectrum [16]–[21], support vector regression [22]–[24], and others [25]–[27]. These algorithms can transparently embed watermark bits into an audio signal and extract the embedded watermark bits from the watermarked audio signal. However, they cannot embed multiple layers of watermark bits into an audio signal.

There are few multi-layer audio watermarking methods reported in the literature [28]–[30]. The method in [28] can embed two sets (or layers) of watermark bits in discrete wavelet transform (DWT) domain. Similarly, the method in [29] can also embed two layers of watermark bits, where one layer of watermark bits are embedded in DWT domain and the other are embedded in time domain. However, these two methods cannot be extended to embed more than two layers of watermark bits. In contrast, the method in [30] can embed several layers of watermark bits, which employs the amplitude modification techniques in [3] to perform watermarking. Since this method embeds watermark bits into an audio signal by directly modifying the amplitude of the samples, it is not robust against high-pass filtering (HPF), echo addition and severe compression attacks. Furthermore, it is very sensitive to time-domain misalignment and thus is also vulnerable to cropping, time-scaling, and pitch-scaling attacks.

In this paper, a new patchwork-based audio watermarking algorithm is proposed to allow watermarking in multiple layers. In the embedding process of our method, the host audio signal is partitioned into audio segments and then the discrete cosine transform (DCT) coefficients of the audio segments are calculated. After discarding those DCT coefficients corresponding to low and high frequency components, the remaining DCT coefficients (corresponding to medium range frequency components) are ordered in a specific manner. This ordering is done to improve the perceptual quality by reducing the amount of modification needed during the watermark embedding process. The ordered DCT coefficients are divided into DCT segments. In order to embed a layer-1 watermark bit, two adjacent DCT segments are considered as a fragment pair. The mean values of each fragment in a fragment pair are modified according to the designed embedding rule. For the layer-2 watermark embedding, four adjacent DCT segments are considered to be a layer-2 fragment pair. A watermark bit is embedded into a layer-2 fragment pair using the same embedding mechanism. Since lengths of the layer-2 fragments are longer than the lengths of the fragments used in layer-1, to maintain imperceptibility and robustness at the desired level, the amount of modification is adjusted in layer-2. Using a similar approach, to embed a watermark bit in layer-N, a layer-N fragment pair is formed using two sets of adjacent 2^N DCT segments. The manner in which DCT coefficients are modified at a particular layer ensures that the watermark bits embedded in the previous layers are not disturbed. At the decoding stage, this feature guarantees the successful extraction of watermark bits from all the layers.

The watermark bits embedded in all the layers using the proposed watermark embedding algorithm are robust to conventional attacks such as noise addition, re-quantization, compression (e.g., MP3 and AAC), filtering (e.g., high-pass and low-pass filtering), and re-sampling attacks. The proposed method is not robust against collusion attacks. However, the watermarks embedded in the higher layers (e.g., production company and main distributors) will be generally safer from collusion attack as it is difficult for an attacker to get different higher layer watermarked copies. Nevertheless, developing a mechanism which can resist collusion attack will be our future work.

The main contributions of the paper can be summarized as follows.

1) A new multi-layer audio watermarking methods is proposed, which allows watermarks to be added in multiple layers in any order.
2) A theoretical analysis is provided to show that adding watermark bits in one layer does not affect the watermark bits embedded in other layers.
3) An error buffer is introduced to increase the robustness against attacks, while using a layer specific error buffer size to reduce perceptual quality degradation.
4) A specifically designed DCT coefficient ordering process is utilized to minimize perceptual quality degradation.

The remainder of the paper is organized as follows. Sections II and III present the embedding and decoding processes of the proposed watermarking algorithm, respectively. The simulation results are shown in Section IV. Section V concludes the paper.

II. Watermark Embedding

A. Generation of DCT Segments

First, the host audio signal is partitioned to obtain a number of segments. Let $x(n)$ be the host audio segment with length L. Denote the DCT coefficients of $x(n)$ by $X(k)$, which is defined as follows [31]:

$$X(k) = l(k) \sum_{n=0}^{L-1} x(n) \cos\left\{ \frac{\pi(2n+1)k}{2L_x} \right\} \tag{1}$$

where $k = 0, 1, \ldots, L-1$, and

$$l(k) = \begin{cases} \frac{1}{\sqrt{L}}, & \text{if } k = 0 \\ \sqrt{\frac{2}{L}}, & \text{if } 1 \le k < L \end{cases}$$

Then, the DCT coefficients corresponding to a certain frequency range $[f_s, f_e]$, denoted by $X_s(k)$, are selected from $X(k)$. Assume that the length of $X_s(k)$ is L_s, is chosen to be an even number. The watermark bits will be embedded into

$X_s(k)$ due to the fact that low and high frequency components are vulnerable to attacks such as filtering and compression.

In audio signals, most of the signal energy is concentrated in the low frequency components. Because of this, generally, $X_s(k)$ takes higher magnitudes for smaller values of k and smaller magnitudes for higher values of k. If we divide $X_s(k)$ into DCT segments, where the ith DCT segment is denoted as $X_{s,i}(k)$, the mean value of the magnitude sequence $|X_{s,i}(k)|$ will vary significantly. To reduce the perceptual quality degradation while embedding watermark bits, the proposed embedding algorithm requires similar means among $|X_{s,i}(k)|$. To achieve this, we rearrange $X_s(k)$ to generate a sequence $X'_s(k)$ of length L_s as follows,

$$\begin{aligned} X'_s(k) = \; & [X_s(0), X_s(L_s-1), X_s(1), X_s(L_s-2), \\ & X_s(2), X_s(L_s-3), \ldots, X_s(L_2/2-1), \\ & X_s(L_s/2)]. \end{aligned} \tag{2}$$

Let us divide $X'_s(k)$ into N_s equal length DCT segments $X'_{s,i}(k)$, $i = 1, 2, \ldots, N_s$, i.e.,

$$X'_{s,i}(k) = [X'_{s,i}(0), X'_{s,i}(1), \ldots, X'_{s,i}(N_s-1)]. \tag{3}$$

The proposed watermarking algorithm embeds watermark bits by altering $E(|X'_{s,i}(k)|)$s, where $E(\cdot)$ stands for averaging operation and $i = 1, 2, \ldots, N_s$. The maximum amount of modification can be reduced when $E(|X'_{s,i}(k)|)$s have similar values. Due to the above mentioned ordering process $E(|X'_{s,i}(k)|)$ will not vary significantly for different values of i. The ordering process of DCT coefficients is illustrated in Fig. 2.

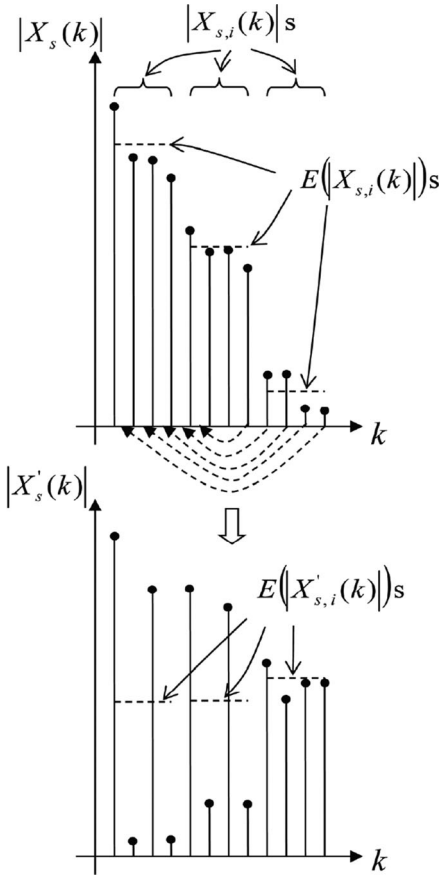

Fig. 2. Ordering process of DCT coefficients.

B. Watermark Embedding Rule

In the proposed method, a watermark bit is embedded into two adjacent groups of rearranged DCT coefficients. We call those two groups as a fragment pair. We denote a layer-j fragment pair as $X^f_{1,j}(k)$ and $X^f_{2,j}(k)$, where $j = 1, 2, \ldots, N$. It is important to note here that there will be several fragment pairs in particular layer. To embed a binary watermark bit in layer-j, $X^f_{1,j}(k)$ and $X^f_{2,j}(k)$ are modified according to the proposed embedding rule.

Let us define the means of the absolute valued fragments as

$$m_{1,j} = E(|X^f_{1,j}(k)|), \tag{4}$$

$$m_{2,j} = E(|X^f_{2,j}(k)|), \tag{5}$$

and we further define the followings terms

$$m_j = \frac{m_{1,j} + m_{2,j}}{2}, \tag{6}$$

$$mm_j = \min\{m_{1,j}, m_{2,j}\}. \tag{7}$$

It is well known that some of the signal processing attacks such as noise addition and re-quantization can slightly modify the mean values. To compensate the errors caused by the change in the mean values, in the embedding rule, we introduce an error buffer to enhance the robustness of the proposed watermarking algorithm. We define a layer-j error buffer parameter as r_j.

Given a fragment pair $X^f_{1,j}(k)$ and $X^f_{2,j}(k)$, a watermark bit is embedded into that fragment pair by modifying $m_{1,j}$ and $m_{2,j}$. Denote the modified counterparts of $m_{1,j}$ and $m_{2,j}$ by $m'_{1,j}$ and $m'_{2,j}$, respectively, the embedding rule is then given as follows.

1) Embedding of watermark bit "0":
   if $(m_{1,j} - m_{2,j}) \geq r_j \cdot mm_j$, then
   $m'_{1,j} = m_{1,j}$,
   $m'_{2,j} = m_{2,j}$.
   Otherwise,
   $m'_{1,j} = m_j + (r_j \cdot mm_j/2)$,
   $m'_{2,j} = m_j - (r_j \cdot mm_j/2)$.
2) Embedding of watermark bit "1":
   if $(m_{2,j} - m_{1,j}) \geq r_j \cdot mm_j$, then
   $m'_{1,j} = m_{1,j}$,
   $m'_{2,j} = m_{2,j}$.
   Otherwise,
   $m'_{1,j} = m_j - (r_j \cdot mm_j/2)$,
   $m'_{2,j} = m_j + (r_j \cdot mm_j/2)$.

After calculating $m'_{1,j}$ and $m'_{2,j}$, the watermark embedded fragments, denoted as $Y^f_{1,j}(k)$ and $Y^f_{2,j}(k)$ respectively, are obtained as follows:

$$Y^f_{1,j} = X^f_{1,j} \times \frac{m'_{1,j}}{m_{1,j}}, \tag{8}$$

$$Y^f_{2,j} = X^f_{2,j} \times \frac{m'_{2,j}}{m_{2,j}}. \tag{9}$$

From (8) and (9), one can clearly see that $E(Y^f_{1,j}) = m'_{1,j}$ and $E(Y^f_{2,j}) = m'_{2,j}$. Similarly, watermark bit can be inserted into all the fragment pairs. Then DCT coefficients in the

watermarked fragments pairs are rearranged back to their original positions. This will be the reverse process of the ordering given in (2). After that, the watermarked audio segments are constructed by using inverse discrete cosine transform (IDCT).

It is noteworthy to mention the significance of the ordering process used in the proposed algorithm. In typical audio signals, the magnitudes of the DCT coefficients dramatically decrease with increasing frequency. Thus, the means of the absolute valued DCT fragments also decrease with the segment number. In the aforementioned embedding rule, depending on the watermark bit, we need to change the mean values of the neighboring absolute valued DCT fragments. Let us assume that the mean value of a first DCT fragment is much higher than that of its fragment pair. To embed a watermark bit, we assume that we need to make the second fragment pair's mean higher than that of the first fragment pair. In this case, we need to modify the DCT coefficients by lager amounts. This will reduce the perceptual quality. On the other hand, the DCT coefficient ordering process facilitates in creating DCT fragment pairs with similar means by considering the frequency varying chrematistics of audio signals. Thus, the ordering process enhances the perceptual quality by reducing the maximum amount of modification required to embed a watermark bit.

The error buffer is used to enhance the robustness of the proposed method against attacks at the expense of reducing imperceptibility. In order to reduce the amount of perceptual quality degradation, the size of the error buffer is adaptively changed. In other words, the adaptive error buffer size improves the perceptual quality compared to the use of a constant buffer size, without notably lowering robustness. To create error buffer, the parameter r_j, where $j = 1, 2, \ldots, N$, is given by

$$r_j = k_1 e^{-k_2 j}, \tag{10}$$

where k_1 and k_2 are positive constants empirically determined. To embed watermark bits in layer-1, fragment pairs $X^f_{1,j}(k)$ and $X^f_{2,j}(k)$, when $j = 1$, are defined as follows:

$$[\{(X'_{s,1}(k)), (X'_{s,2}(k))\}, \{(X'_{s,3}(k)), (X'_{s,4}(k))\}, \ldots].$$

Here $[\{(a_1), (a_2)\}, \{(b_1), (b_2)\}]$ denotes the fragment pairs a_1, a_2 and b_1, b_2. Moreover, in this context $(\cdot)$ denotes a fragment and $\{(\cdot), (\cdot)\}$ represents a fragment pair. If we consider layer-1 first fragment pair, then $X^f_{1,1}(k) = X'_{s,1}(k)$ and $X^f_{2,1}(k) = X'_{s,2}(k)$. Similarly, layer-1 second fragment pair is $\{(X'_{s,3}(k)), (X'_{s,4}(k))\}$. The layer-1 error buffer related parameter $r_1 = k_1 e^{-k_2 \times 1}$. Let us represent the layer-1 watermarked counterparts of $X'_{s,i}(k)$s by $Y^1_{s,i}(k)$s, respectively. To embed watermark bits in layer-2, fragment pairs $X^f_{1,j}(k)$ and $X^f_{2,j}(k)$, when $j = 2$, are defined as follows:

$$[\{(Y^1_{s,1}(k), Y^1_{s,2}(k)), (Y^1_{s,3}(k), Y^1_{s,4}(k))\},$$
$$\{(Y^1_{s,5}(k), Y^1_{s,6}(k)), (Y^1_{s,7}(k), Y^1_{s,8}(k))\}, \ldots].$$

If we consider layer-2 first fragment pair, $X^f_{1,2}(k) = (Y^1_{s,1}(k), Y^1_{s,2}(k))$ and $X^f_{2,2}(k) = (Y^1_{s,3}(k), Y^1_{s,4}(k))$. The layer-2 error buffer related parameter $r_2 = k_1 e^{-k_2 \times 2}$. We denote the layer-2 watermarked counterparts of $Y^1_{s,i}(k)$ by $Y^2_{s,i}(k)$, respectively. It should be noted here that instead of $Y^1_{s,i}(k)$, the unwatermarked counterparts $X'_{s,i}(k)$ can also be used for layer-2 watermarking. In the similar manner, to embed watermark bits in layer-N, fragment pairs $X^f_{1,j}(k)$ and $X^f_{2,j}(k)$, when $j = N$, are defined as follows:

$$[\{(Y^{N-1}_{s,1}(k), Y^{N-1}_{s,2}(k), \ldots, Y^{N-1}_{s,2^{N-1}}(k)),$$
$$(Y^{N-1}_{s,2^{N-1}+1}(k), Y^{N-1}_{s,2^{N-1}+2}(k), \ldots, Y^{N-1}_{s,2\times 2^{N-1}}(k))\},$$
$$\{(Y^{N-1}_{s,2^N+1}(k), Y^{N-1}_{s,2^N+2}(k), \ldots, Y^{N-1}_{s,3\times 2^{N-1}}(k)),$$
$$(Y^{N-1}_{s,3\times 2^{N-1}+1}(k), Y^{N-1}_{s,3\times 2^{N-1}+2}(k), \ldots, Y^{N-1}_{s,4\times 2^{N-1}}(k))\}, \ldots].$$

If we consider layer-N first fragment pair, $X^f_{1,N}(k) = (Y^{N-1}_{s,1}(k), Y^{N-1}_{s,2}(k), \ldots, Y^{N-1}_{s,2^{N-1}}(k))$ and $X^f_{2,N}(k) = (Y^{N-1}_{s,2^{N-1}+1}(k), Y^{N-1}_{s,2^{N-1}+2}(k), \ldots, Y^{N-1}_{s,2\times 2^{N-1}}(k))$. The layer-$N$ error buffer related parameter $r_N = k_1 e^{-k_2 \times N}$. Let us represent the layer-N watermarked counterparts of $Y^{N-1}_{s,1}(k)$s by $Y^N_{s,1}(k)$s, respectively. Fig. 3 depicts sample fragment pairs of first three layers.

It is noteworthy to mention here that there is no need to predetermine the number of layers (N) at the beginning. One can see from the fragment pair generation process that we can form fragment pairs corresponding to a new layer at a later stage if necessary.

The embedding rule is designed in such a way that the watermark bits embedded in one layer will not affect the watermark bits embedded in the previous layers. This will be explained clearly in the next section. It is worthwhile to mention here that the length of the fragment pairs increases with increased layer index. Thus, to maintain the amount of modification, size of the error buffer is reduced by reducing r_j with increasing layer index j.

III. Watermark Decoding

At the decoding end, we use the segmenting procedure utilized in the embedding process to partition the multi-layer watermarked audio signal. Then, we obtain the DCT coefficients of those segments. Let us denote watermark embedded potion of the DCT coefficients by $Y_s(k)$. After that using the similar ordering process utilized in (2) at the embedding stage, we generate $Y'_s(k)$. Then $Y'_s(k)$ is divided into N_s equal length DCT segments $Y'_{s,i}(k)$, $i = 1, 2, \ldots, N_s$.

A. Decoding Rule

In an attack free, i.e., closed-loop, environment, we can generate watermarked layer-j fragment pair $Y^f_{1,j}(k)$ and $Y^f_{2,j}(k)$, where $j = 1, 2, \ldots, N$. Let us define means of the absolute valued watermarked fragments as follows:

$$m'_{1,j} = E(|Y^f_{1,j}(k)|), \tag{11}$$

$$m'_{2,j} = E(|Y^f_{2,j}(k)|). \tag{12}$$

Using the two mean values, we calculate their difference,

$$d_j = m'_{1,j} - m'_{2,j}. \tag{13}$$

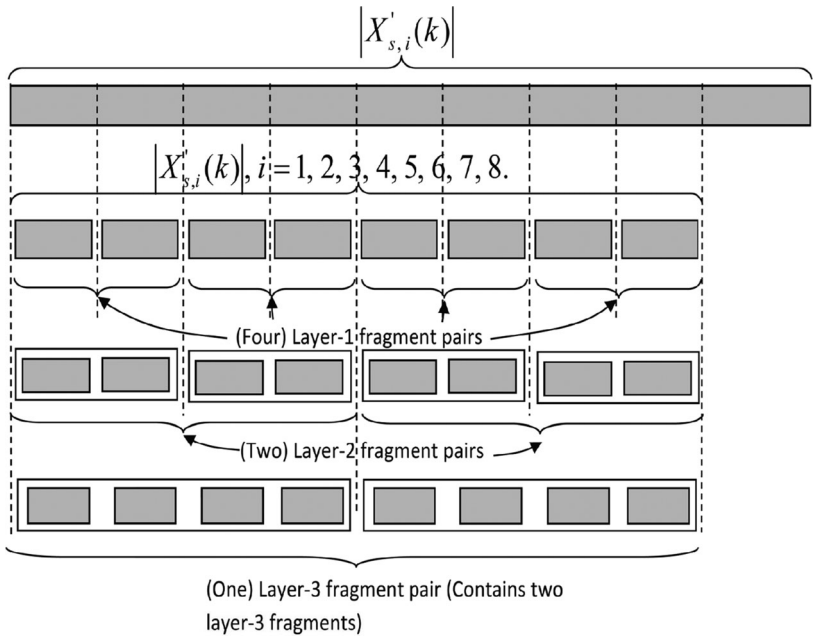

Fig. 3. Fragment pairs of first three layers.

The watermark extraction decision criterion is as follows. Watermark bit "0" is extracted from a watermarked fragment pair if $d_j \geq 0$. Otherwise, watermark bit "1" is extracted from the watermarked fragment pair.

Without loss of generality, let us assume the watermarked fragment pair $Y^f_{1,j}(k)$ and $Y^f_{2,j}(k)$ contain the watermark bit "0". Then, based on the embedding rule described in the previous section, from (13), we can easily derive

$$\begin{aligned} d_j &= m'_{1,j} - m'_{2,j} \\ &\geq r_j \cdot mm_j. \end{aligned} \tag{14}$$

Similarly, when the embedded watermark bit is "1", we can show that $d_j \leq -r_j \cdot mm_j$. Assume that due to the impact of an attack on the watermarked signal, d_j becomes $d_j + \epsilon$, where ϵ is an error. According to the aforementioned watermark extraction decision criterion, to successfully extract watermark bit "0" (respectively watermark bit "1"), we only require $d_j \geq 0$ (respectively $d_j < 0$). Thus, as long as $abs(\epsilon) < r_j \cdot mm_j$, the embedded watermark bit "0" (respectively the embedded watermark bit "1") can be extracted correctly. This means that $r_j \cdot mm_j$ provides an error buffer to tolerate the impact of attacks on watermark extraction, where r_j is referred as error buffer parameter.

When it comes to robustness against common attacks, the proposed method can withstand attacks such as noise addition and re-quantization due to the introduction of the error buffer. Furthermore, the proposed algorithm can resist amplification attack, as this attack does not disturb the relationship between the mean values of the magnitudes of the fragment pairs. In the case of compression (e.g., MP3 and AAC compression) and filtering (e.g., high-pass and low-pass filtering) attacks low and high frequency components of the audio signals are severely affected. Since the proposed method does not utilize those frequency regions to embed watermark bits (regardless of the layer of watermarking), the proposed method is also robust against the compression and filtering attacks.

It is important to note here that the error buffer size is also dependent on mm_j, which is the smallest mean value of the magnitudes of the two fragments. This implies that the error buffer size is inversely related to the strength of the weakest fragment in the fragment pair. Thus, the selection of error buffer size $r_j \cdot mm_j$ reduces the perceptual quality degradation in multi-layer watermark bit embedding.

B. Multi-Layer Watermark Extraction

The primary objective of the proposed watermarking algorithm is to transparently embed watermark bits into multiple layers and successfully extract the watermark bits from all the embedded layers. In the proposed watermarking method, the embedded watermark bits in a particular layer are extracted by comparing the absolute valued means of fragment pairs in that layer. The watermark embedding algorithm is designed in such a way that the relationship between the absolute valued means of fragment pairs belonging to a layer is not disturbed by the watermark embedding in other layers.

To understand how the proposed watermarking algorithm handles the multi-layer watermarking, without loss of generality, let us consider a two layer watermarking scenario and the watermark bits embedded in layer-1 are "0"s and watermark bits embedded in layer-2 are "1"s. We also assume that the Layer-1 first fragment pair is $\{(X'_{s,1}(k)), (X'_{s,2}(k))\}$ and second fragment pair is $\{(X'_{s,3}(k)), (X'_{s,4}(k))\}$. Thus, the watermarked counterparts of these fragment pairs will be $\{(Y^1_{s,1}(k)), (Y^1_{s,2}(k))\}$ and $\{(Y^1_{s,3}(k)), (Y^1_{s,4}(k))\}$. Then, for layer-2 watermark embedding the layer-2 fragment pair will be $\{(Y^1_{s,1}(k), Y^1_{s,2}(k)), (Y^1_{s,3}(k), Y^1_{s,4}(k))\}$ and its watermarked fragment pair will be $\{(Y^2_{s,1}(k), Y^2_{s,2}(k)), (Y^2_{s,3}(k), Y^2_{s,4}(k))\}$. Now, we consider two cases.

1) *Case 1:* Let us assume that 2 watermark bits of "0" are embedded into 2 layer-1 fragment pairs. Then watermark bit "1" is embedded into a layer-2 fragment pair.
Since "0"s are embedded in layer-1 fragment pairs, based on the embedding rule, we have

$$E\left(|Y^1_{s,1}(k)|\right) > E\left(|Y^1_{s,2}(k)|\right), \tag{15}$$

$$E\left(|Y^1_{s,3}(k)|\right) > E\left(|Y^1_{s,4}(k)|\right). \tag{16}$$

As watermark bit "1" is embedded into the layer-2 fragment pair, we can write

$$E\left(|[Y^2_{s,1}(k), Y^2_{s,2}(k)]|\right) < E\left(|[Y^2_{s,3}(k), Y^2_{s,4}(k)]|\right) \tag{17}$$

Since layer-2 watermarks are added in the last step, without any problem, based on (17), we can extract the embedded watermark bit "1". Now, let us consider the effects caused by layer-2 modification on the layer-1 watermarked fragments $Y^1_{s,i}(k)$s where $i = 1, 2, 3, 4$. According to (8) and (9), because of the layer-2 watermarking $Y^1_{s,i}(k)$ becomes $Y^2_{s,i}(k)$ which are given as follows,

$$Y^2_{s,1}(k) = \alpha_1 Y^1_{s,1}(k), \tag{18}$$

$$Y^2_{s,2}(k) = \alpha_1 Y^1_{s,2}(k), \tag{19}$$

$$Y^2_{s,3}(k) = \alpha_2 Y^1_{s,3}(k), \tag{20}$$

$$Y^2_{s,4}(k) = \alpha_2 Y^1_{s,4}(k), \tag{21}$$

where the positive values α_1 and α_2 can be computed using (8) and (9). Since layer-2 watermarking does not change the relationship between the absolute valued means of layer-1 fragments in a fragment pair, we have

$$E\left(|Y^2_{s,1}(k)|\right) > E\left(|Y^2_{s,2}(k)|\right), \tag{22}$$

$$E\left(|Y^2_{s,3}(k)|\right) > E\left(|Y^2_{s,4}(k)|\right). \tag{23}$$

From the inequalities (22) and (23) we can successfully extract the embedded watermark bits in layer-1 according to the extraction rule.

2) *Case 2:* Let us assume a different embedding order from Case 1, where watermark bit "1" is embedded into a layer-2 fragment pair. After that, 2 watermark bits of "0" are embedded into 2 layer-1 fragment pairs.
Similar to the previous case, for layer-2 fragment pair, we can have

$$E\left(|[Y^2_{s,1}(k), Y^2_{s,2}(k)]|\right) < E\left(|[Y^2_{s,3}(k), Y^2_{s,4}(k)]|\right) \tag{24}$$

In a similar manner, for layer-1 fragment pairs, we have

$$E\left(|Y^1_{s,1}(k)|\right) > E\left(|Y^1_{s,2}(k)|\right), \tag{25}$$

$$E\left(|Y^1_{s,3}(k)|\right) > E\left(|Y^1_{s,4}(k)|\right). \tag{26}$$

Since layer-1 watermarks are added in the last step, without any difficulty, based on (17), we can successfully extract both embedded watermark bits (i.e., two "0" bits here). Now, we consider the effect caused by layer-1 watermark embedding on layer-2 watermarked fragments $[Y^1_{s,1}(k), Y^1_{s,2}(k)]$ and $[Y^1_{s,3}(k), Y^1_{s,4}(k)]$. Let us denote the layer-2 watermarked fragments $[Y^2_{s,1}(k), Y^2_{s,2}(k)]$ and $[Y^2_{s,3}(k), Y^2_{s,4}(k)]$ altered by the layer-1 watermark embedding as $[Y^1_{s,1}(k), Y^1_{s,2}(k)]$ and $[Y^1_{s,3}(k), Y^1_{s,4}(k)]$, respectively. Then we can write,

$$\left[Y^1_{s,1}(k), Y^1_{s,2}(k)\right] = \left[\beta_1 Y^2_{s,1}(k), \beta_2 Y^2_{s,2}(k)\right] \tag{27}$$

$$\left[Y^1_{s,3}(k), Y^1_{s,4}(k)\right] = \left[\beta_3 Y^2_{s,3}(k), \beta_4 Y^2_{s,4}(k)\right] \tag{28}$$

where $\beta_1, \beta_2, \beta_3$ and β_4 are some positive numbers. In this case, to successfully extract the embedded watermark bit from layer-2, we expect the following inequality to hold,

$$E\left(|[Y^1_{s,1}(k), Y^1_{s,2}(k)]|\right) < E\left(|[Y^1_{s,3}(k), Y^1_{s,4}(k)]|\right) \tag{29}$$

First, we consider $\left[\beta_1 Y^2_{s,1}(k), \beta_2 Y^2_{s,2}(k)\right]$. Since $Y^2_{s,1}(k)$ and $Y^2_{s,1}(k)$ are of same length, we can derive

$$\begin{aligned} &E\left(|\left[\beta_1 Y^2_{s,1}(k), \beta_2 Y^2_{s,2}(k)\right]|\right) \\ &= \frac{E\left(|\beta_1 Y^2_{s,1}(k)|\right) + E\left(|\beta_2 Y^2_{s,2}(k)|\right)}{2} \\ &= \frac{\beta_1 E\left(|Y^2_{s,1}(k)|\right) + \beta_2 E\left(|Y^2_{s,2}(k)|\right)}{2}. \end{aligned} \tag{30}$$

By considering (8) and (9), in this instance, we can write

$$m_{1,j} = E\left(|Y^2_{s,1}(k)|\right), \tag{31}$$

$$m_{2,j} = E\left(|Y^2_{s,2}(k)|\right), \tag{32}$$

$$\beta_1 = \frac{m'_{1,j}}{m_{1,j}}, \tag{33}$$

$$\beta_2 = \frac{m'_{2,j}}{m_{2,j}}. \tag{34}$$

From (31)–(34), we have

$$\begin{aligned} \beta_1 E\left(|Y^2_{s,1}(k)|\right) &= \beta_1 \times m_{1,j} \\ &= \frac{m'_{1,j}}{m_{1,j}} \times m_{1,j} \\ &= m'_{1,j}, \end{aligned} \tag{35}$$

$$\begin{aligned} \beta_2 E\left(|Y^2_{s,2}(k)|\right) &= \beta_2 \times m_{2,j} \\ &= \frac{m'_{2,j}}{m_{2,j}} \times m_{2,j} \\ &= m'_{2,j}. \end{aligned} \tag{36}$$

Substituting (35) and (36) into (30) yields

$$E\left(|\left[\beta_1 Y^2_{s,1}(k), \beta_2 Y^2_{s,2}(k)\right]|\right) = \frac{m'_{1,j} + m'_{2,j}}{2}. \tag{37}$$

Based on the embedding algorithm, we can observe that

$$\frac{m'_{1,j} + m'_{2,j}}{2} = \frac{m_{1,j} + m_{2,j}}{2}.$$

TABLE I
HOST AUDIO SIGNALS USED IN SIMULATIONS

Group no.	Host signals	Genres
1	$S_{01} \sim S_{10}$	Western rock, pop & roll music
2	$S_{11} \sim S_{20}$	Eastern classical & folk music
3	$S_{21} \sim S_{30}$	Speeches (Male & Female voices)
4	$S_{31} \sim S_{40}$	Nature (Rain & wind sounds)

Then, according to (31), (32), and (37), we have

$$E\left(\left|\left[\beta_1 Y^2_{s,1}(k), \beta_2 Y^2_{s,2}(k)\right]\right|\right) = \frac{E\left(\left|Y^2_{s,1}(k)\right|\right) + E\left(\left|Y^2_{s,2}(k)\right|\right)}{2} = E\left(\left|\left[Y^2_{s,1}(k), Y^2_{s,2}(k)\right]\right|\right). \quad (38)$$

Similarly, we can obtain the following equality

$$E\left(\left|\left[\beta_3 Y^2_{s,3}(k), \beta_4 Y^2_{s,4}(k)\right]\right|\right) = E\left(\left|\left[Y^2_{s,3}(k), Y^2_{s,4}(k)\right]\right|\right). \quad (39)$$

From (39), (38), and (24), it follows

$$E\left(\left|\left[Y^2_{s,1}(k), Y^2_{s,2}(k)\right]\right|\right) < E\left(\left|\left[Y^2_{s,3}(k), Y^2_{s,4}(k)\right]\right|\right),$$

$$E\left(\left|\left[\beta_1 Y^2_{s,1}(k), \beta_2 Y^2_{s,2}(k)\right]\right|\right) < E\left(\left|\left[\beta_3 Y^2_{s,3}(k), \beta_4 Y^2_{s,4}(k)\right]\right|\right). \quad (40)$$

Substituting (27) and (28) in (40) yields

$$E\left(\left|\left[Y^1_{s,1}(k), Y^1_{s,2}(k)\right]\right|\right) < E\left(\left|\left[Y^1_{s,3}(k), Y^1_{s,4}(k)\right]\right|\right) \quad (41)$$

From (41) we can successfully extract the layer-2 watermark bit "1" from the fragment pair.

This clearly shows that when proposed algorithm is used for watermarking, the watermark bits embedded in one layer do not affect the watermarks in other layers. Due to this feature, we can extract the watermark bits from a multi-layer watermarked audio signal without any errors.

IV. SIMULATION RESULTS

In this section, the proposed multi-layer watermarking algorithm's perceptual quality and the robustness against common attacks are illustrated by simulation results. In the simulations, 40 randomly selected mono-channel audio signals belonging to four different genres (see Table I)are used as host audio signals. Each audio clip has a duration of 10 seconds, and all of them are sampled at the rate of 44.1 kHz and quantized with 16-bit depth. The DCT coefficients are calculated using all the sample values and those DCT coefficients corresponding to frequency range 3 kHz to 7 kHz are used for watermark embedding (i.e., $f_s = 3$ kHz and $f_e = 7$ kHz). We set $L = 17640$, $L_s = 3200$, $N_s = 800$, $k_1 = 0.195$, and $k_2 = 0.08$. The default number of layers is $N = 3$, unless mentioned otherwise. We have empirically chosen these parameters by considering robustness, perceptual quality, and the embedding capacity.

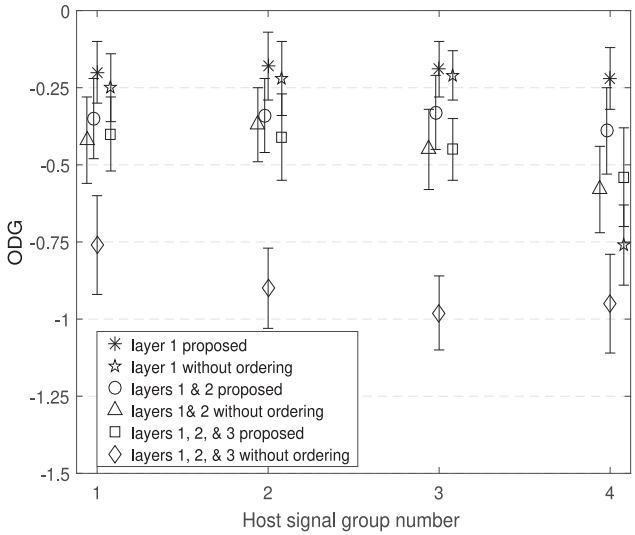

Fig. 4. Perceptual quality (in terms of ODGs) under $N = 1, 2, 3$, versus host audio signal group number, with and without ordering DCT coefficients (N = 1: Only layer-1 is watermarked, N = 2: Layers 1 and 2 are watermarked, N = 3: Layers 1, 2, and 3 are watermarked).

Based on the selected parameters, the embedding capacity of the proposed method's layer-1, layer-2 and layer-3 can be computed as 10 bps, 5 bps and 2.5 bps, respectively. Based on the application requirements, the embedding capacity can be increased by reducing the segment length, at the expense of robustness and perceptual quality.

Since robustness and the perceptual quality are inversely related, we need to access the perceptual quality of audio signals when they are watermarked using the proposed algorithm. To assess the perceptual quality we employ the perceptual evaluation of audio quality (PEAQ) algorithm [32], [33], as used in [4] and [34]. The PEAQ algorithm returns a parameter called objective difference grade (ODG) which is between -4 and 0, in which 0 corresponds to imperceptible while 4 means annoying artifacts (lowest quality). The perceptual quality increases with the ODG value.

Maintaining acceptable perceptual quality in multi-layer audio watermarking is a daunting task. To enhance the perceptual quality of the watermarked audio signal, the proposed watermarking algorithm employs a special DCT coefficient ordering process which is given in (2). To analyze the effect of this ordering, in Fig. 4, ODG values are compared for proposed algorithm with and without DCT coefficient ordering process, against various audio signal groups. Across all the signal groups, from Fig. 4 we can see that the algorithm which uses the ordered DCT coefficients exhibits better perceptual quality. From Fig. 4, it should also be noted that when all the three layers are watermarked, DCT coefficient ordering process significantly improves the perceptual quality. Since perceptual quality is inversely related to the robustness it is important to analyze the effects caused by DCT coefficient ordering process in the robustness. In Table II, we compared robustness of the proposed algorithm with and without the ordering process. In this comparison, we only consider attacks where the proposed algorithm (with ordering) cannot achieve DRs of 100%. As the proposed method gets DRs of

TABLE II
DETECTION RATES OF THE PROPOSED SCHEME WITH AND WITHOUT ORDERING THE DCT COEFFICIENTS

Attacks	Host signals	Proposed method's DR_1s (%)	
		Without ordering	With ordering
Noise	$S_{01} \sim S_{10}$	99.6	99.9
	$S_{11} \sim S_{20}$	99.8	100
	$S_{21} \sim S_{30}$	99.7	100
	$S_{31} \sim S_{40}$	99.7	99.7
MP3 (64 kbps)	$S_{01} \sim S_{10}$	100	100
	$S_{11} \sim S_{20}$	100	100
	$S_{21} \sim S_{30}$	100	100
	$S_{31} \sim S_{40}$	99.6	99.6
AAC (64 kbps)	$S_{01} \sim S_{10}$	100	100
	$S_{11} \sim S_{20}$	100	100
	$S_{21} \sim S_{30}$	100	100
	$S_{31} \sim S_{40}$	99.5	99.6

100% in layer 2 and 3, only layer-1 is considered in this comparison. We can observe from Table II that DRs are similar for proposed algorithm with and without the ordering process. However, the proposed algorithm with ordering process is slightly better than its un-ordered counterpart. This shows that the DCT coefficient ordering process increases the perceptual quality of the proposed algorithm without compromising the robustness.

To objectively measure the robustness of the proposed watermarking algorithm, the detection rate (DR) is used. We define detection rate DR_j

$$\mathrm{DR} = \left(\frac{\text{Number of watermarks correctly extracted}}{\text{Number of watermarks embedded}}\right) \times 100\%.$$

The following common attacks are used in the evaluation of robustness:

1) *Closed-loop attack (CL)*: The watermarks are extracted from the watermarked signals in an attack-free environment.
2) *Re-quantization attack (RQ)*: Each sample of the watermarked signals is re-quantized from 16-bits to 8-bit quantization depth.
3) *Noise attack (Noise)*: Random noise is added to the watermarked signals, where the signal-to-noise ratio (SNR) is 20 dB.
4) *Amplitude attack (Amp)*: The amplitudes of the watermarked signals are enlarged, where the amplifications are 1.2 and 1.5 times.
5) MP3 *attack*: MPEG-1 Layer-III compression is performed on the watermarked signals, where the compression bit rates are 128 kbps, 96 kbps, and 64 kbps, respectively.
6) AAC *attack*: MPEG-4 advanced audio coding based compression is performed on the watermarked signals, where the compression bit rates are 128 kbps, 96 kbps, and 64 kbps, respectively.

TABLE III
ODG VALUES OF DIFFERENT SIGNALS VERSUS DIFFERENT NUMBER OF LAYERS

Host signals	Only layer-1 is watermarked ($N = 1$)		Layers 1 and 2 are watermarked ($N = 2$)		All 3 layers are watermarked ($N = 3$)	
	ODG	std	ODG	std	ODG	std
$S_{01} \sim S_{10}$	-0.19	0.10	-0.35	0.13	-0.40	0.12
$S_{11} \sim S_{20}$	-0.18	0.11	-0.34	0.12	-0.41	0.14
$S_{21} \sim S_{30}$	-0.19	0.09	-0.32	0.12	-0.45	0.10
$S_{31} \sim S_{40}$	-0.21	0.10	-0.39	0.14	-0.54	0.16

7) *High-pass filtering (HPF) attack*: High-pass filter is applied to the watermarked signals, where the cut-off frequency is 1 kHz.
8) *Low-pass filtering (LPF) attack*: Low-pass filter is applied to the watermarked signals, where the cut-off frequency is 8 kHz.
9) *Echo addition*: The echo signal with the scaling factor of 0.2 and the time delay of 0.5 s is added to the watermarked signals.
10) *Re-sampling attack (RS)*: The watermarked signals are down-sampled to 16 kHz and then up-sampled back to 44.1 kHz (i.e., 44.1 kHz $\rightarrow$ 16 kHz $\rightarrow$ 44.1 kHz).
11) *Cropping attack*: From the watermarked audio signals first 6000 samples are removed.
12) *Time-scaling attack*: The watermarked audio signals are time-scaled by +1% (i.e., Speed up by 1%) and -1% (i.e., Slow down by 1%) .
13) *Pitch-scaling attack*: The watermarked audio signals are pitch-scaled by +1% (i.e., Pitch is lowered by 1%) and -1% (i.e., Pitch is raised by 1%).

First, to understand the effect of multi-layer watermarking in the perceptual quality, we compare the ODG values in the following cases.

1) *Case 1:* When watermark bits are only embedded into layer-1 (i.e., $N = 1$).
2) *Case 2:* When watermark bits are embedded into layers-1 and 2 (i.e., $N = 2$).
3) *Case 3:* When watermark bits are embedded into layers-1, 2, and 3 (i.e., $N = 3$).

The ODG values corresponding to different types of audio signals with standard deviations are shown in Table III. From the table, it is clear that, as expected, ODG values decrease when more layer of watermark bits are added (i.e., ODG value decreases with increasing j). However, we can see from the table that all ODG values are greater than -0.6. The ODG value greater than -0.6 indicates the high perceptual quality of the watermarking algorithm.

We evaluated the robustness of the proposed method and the amplitude modification based multi-layer watermarking method in [30]. We compare both approaches under the same perceptual quality with ODG $= -0.6$. Due to the nature of the algorithm the proposed methods embedding capacity cannot be freely adjusted. Thus, we kept layer 1, 2, and 3 embedding capacities of the method in [30] at 9.9 bps, 3.3 bps, and

TABLE IV
DETECTION RATES OF THE PROPOSED SCHEME ($N = 3$) AND THE METHOD IN [30] AGAINST ATTACKS, CORRESPONDING TO ALL THREE LAYERS, WHERE ODG $= -0.6$ FOR BOTH METHODS

Attacks	Host signals	DRs (%)					
		Layer 1		Layer 2		Layer 3	
		Method in [30]	Proposed	Method in [30]	Proposed	Method in [30]	Proposed
Closed-loop	$S_{01} \sim S_{10}$	100	100	100	100	100	100
	$S_{11} \sim S_{20}$	100	100	100	100	100	100
	$S_{21} \sim S_{30}$	100	100	100	100	100	100
	$S_{31} \sim S_{40}$	100	100	100	100	100	100
Re-quantization	$S_{01} \sim S_{10}$	100	100	100	100	100	100
	$S_{11} \sim S_{20}$	99.7	100	100	100	100	100
	$S_{21} \sim S_{30}$	100	100	100	100	100	100
	$S_{31} \sim S_{40}$	99.2	100	100	100	100	100
Noise	$S_{01} \sim S_{10}$	97.5	99.9	100	100	100	100
	$S_{11} \sim S_{20}$	97.1	100	100	100	100	100
	$S_{21} \sim S_{30}$	98.2	99.9	98.6	100	98.8	100
	$S_{31} \sim S_{40}$	98.3	99.7	99.2	100	99.8	100
Amplitude (1.2)	$S_{01} \sim S_{10}$	100	100	100	100	100	100
	$S_{11} \sim S_{20}$	100	100	100	100	100	100
	$S_{21} \sim S_{30}$	100	100	100	100	100	100
	$S_{31} \sim S_{40}$	100	100	100	100	100	100
Amplitude (1.5)	$S_{01} \sim S_{10}$	100	100	100	100	100	100
	$S_{11} \sim S_{20}$	100	100	100	100	100	100
	$S_{21} \sim S_{30}$	100	100	100	100	100	100
	$S_{31} \sim S_{40}$	100	100	100	100	100	100
HPF (1 kHz)	$S_{01} \sim S_{10}$	61.3	100	53.2	100	54.6	100
	$S_{11} \sim S_{20}$	52.0	100	47.7	100	54.5	100
	$S_{21} \sim S_{30}$	57.3	100	62.1	100	56.8	100
	$S_{31} \sim S_{40}$	53.5	100	46.2	100	54.5	100
LPF (8 kHz)	$S_{01} \sim S_{10}$	99.4	100	100	100	100	100
	$S_{11} \sim S_{20}$	89.3	100	90.1	100	95.4	100
	$S_{21} \sim S_{30}$	99.6	100	98.1	100	99.7	100
	$S_{31} \sim S_{40}$	99.2	100	100	100	99.9	100
Echo addition	$S_{01} \sim S_{10}$	74.8	100	88.6	100	84.1	100
	$S_{11} \sim S_{20}$	76.0	100	87.9	100	95.5	100
	$S_{21} \sim S_{30}$	70.2	100	75.8	99.9	70.5	99.7
	$S_{31} \sim S_{40}$	78.5	99.9	84.8	98.7	93.2	97.6
Re-sampling	$S_{01} \sim S_{10}$	99.4	100	100	100	100	100
	$S_{11} \sim S_{20}$	90.6	100	91.7	100	97.7	100
	$S_{21} \sim S_{30}$	90.4	100	90.9	100	97.5	100
	$S_{31} \sim S_{40}$	99.2	100	99.4	100	100	100

1.1 bps, respectively. This means that the proposed method's embedding capacities are maintained higher than that ofthe method in [30].Tables IV–VI show the DRs of the proposed method and the method in [30] from all the three layers, against common attacks. For all the simulation outcomes presented in Tables IV–VI, watermark embedding is performed in all the three layers.

It can be seen from Table IV that the proposed method consistently outperforms the method in [30] under all considered attacks across all the layers. For the attacks considered in Table IV the proposed algorithm can successfully extract all the watermark bits from all the three layers against all the attacks except noise addition and echo addition attacks. Table V summarizes the DRs of the proposed watermarking algorithm and the method in [30] under MP3 and AAC attacks at different compression bit rates: 128 kbps, 96 kbps, and 64 kbps. These compression bit rates are commonly used in practical applications. We can see from Table V that under both compression attacks, at compression bit rates 128 kbps and 96 kbps, the proposed algorithm can extract watermark bits from all types of audio signals without any errors. Table VI shows the DRs of the proposed method and the method in [30] against desynchronization attacks. In cropping, time-scaling, and pitch-scaling attacks the proposed method outperforms the method in [30] by a large margin. For most of these desynchronization attacks the method in [30] achieves DRs closer to the chance level of 50%. From the simulation results it is clear that the proposed watermarking algorithm consistently performs well across different types of audio signals and attacks.

Furthermore, we also compared the proposed method with the popular patchwork based method in [4]. As the method in [4] is not designed to support multi-layer watermarking, we compared the proposed method's layer-1 DRs with the DRs of the method in [4]. We compared both methods under same

TABLE V
DETECTION RATES OF THE PROPOSED SCHEME ($N = 3$) AND THE METHOD IN [30] AGAINST COMPRESSION ATTACKS, CORRESPONDING TO ALL THREE LAYERS, WHERE ODG $= -0.6$ FOR BOTH METHODS

Attacks	Host signals	DRs (%)					
		Layer 1		Layer 2		Layer 3	
		Method in [30]	Proposed	Method in [30]	Proposed	Method in [30]	Proposed
MP3 (128 kbps)	$S_{01} \sim S_{10}$	98.4	100	98.8	100	100	100
	$S_{11} \sim S_{20}$	90.1	100	93.9	100	97.5	100
	$S_{21} \sim S_{30}$	91.9	100	91.4	100	82.3	100
	$S_{31} \sim S_{40}$	98.4	100	99.7	100	99.9	100
MP3 (96 kbps)	$S_{01} \sim S_{10}$	96.4	100	97.7	100	100	100
	$S_{11} \sim S_{20}$	92.8	100	93.1	100	95.4	100
	$S_{21} \sim S_{30}$	87.8	100	90.8	100	81.0	100
	$S_{31} \sim S_{40}$	97.4	100	99.1	100	99.8	100
MP3 (64 kbps)	$S_{01} \sim S_{10}$	93.4	100	94.7	100	100	100
	$S_{11} \sim S_{20}$	92.5	100	93.1	100	95.2	100
	$S_{21} \sim S_{30}$	86.2	100	86.8	100	87.5	100
	$S_{31} \sim S_{40}$	97.1	99.6	98.5	100	99.8	100
AAC (128 kbps)	$S_{01} \sim S_{10}$	98.2	100	98.8	100	100	100
	$S_{11} \sim S_{20}$	89.4	100	93.5	100	97.1	100
	$S_{21} \sim S_{30}$	91.3	100	91.1	100	89.2	100
	$S_{31} \sim S_{40}$	98.1	100	99.1	100	98.7	100
AAC (96 kbps)	$S_{01} \sim S_{10}$	95.9	100	97.3	100	100	100
	$S_{11} \sim S_{20}$	89.1	100	92.4	100	95.3	100
	$S_{21} \sim S_{30}$	90.8	100	91.0	100	88.7	100
	$S_{31} \sim S_{40}$	97.3	100	98.2	100	98.6	100
AAC (64 kbps)	$S_{01} \sim S_{10}$	93.1	100	94.4	100	99.9	100
	$S_{11} \sim S_{20}$	89.0	100	92.1	100	94.8	100
	$S_{21} \sim S_{30}$	90.2	100	89.4	100	88.3	100
	$S_{31} \sim S_{40}$	97.1	99.5	98.2	100	98.5	100

TABLE VI
DETECTION RATES OF THE PROPOSED SCHEME ($N = 3$) AND THE METHOD IN [30] AGAINST DESYNCHRONIZATION ATTACKS, CORRESPONDING TO ALL THREE LAYERS, WHERE ODG $= -0.6$ FOR BOTH METHODS

Attacks	Host signals	DRs (%)					
		Layer 1		Layer 2		Layer 3	
		Method in [30]	Proposed	Method in [30]	Proposed	Method in [30]	Proposed
Cropping	$S_{01} \sim S_{10}$	51.2	99.7	52.9	99.9	49.7	99.1
	$S_{11} \sim S_{20}$	48	99.9	50.1	99.9	43.2	98.2
	$S_{21} \sim S_{30}$	43.9	98.7	55.3	98.1	47.8	97.6
	$S_{31} \sim S_{40}$	42.2	99.4	53	99.8	59.1	98.5
Time-scaling (+1%)	$S_{01} \sim S_{10}$	52.4	99.9	46.8	98.5	48.1	60.2
	$S_{11} \sim S_{20}$	48.7	100	44.7	98.8	59.1	58.4
	$S_{21} \sim S_{30}$	45.2	99.5	51.4	99.2	52.4	61.4
	$S_{31} \sim S_{40}$	48.5	97.3	47.7	97.8	72.7	57.1
Time-scaling (-1%)	$S_{01} \sim S_{10}$	43.9	99.4	51.8	99.1	49.7	55.8
	$S_{11} \sim S_{20}$	51	99.0	45.5	98.7	52.3	59
	$S_{21} \sim S_{30}$	52.3	99.8	47.7	99.6	56.8	60.4
	$S_{31} \sim S_{40}$	54.3	100	49.2	99.2	54.5	58.2
Pitch-scaling (+1%)	$S_{01} \sim S_{10}$	61.4	97.7	85.1	99.2	99.6	100
	$S_{11} \sim S_{20}$	73.9	93.2	78.8	99.1	96.7	99.9
	$S_{21} \sim S_{30}$	65.2	94.1	73.5	98.2	80.6	99.7
	$S_{31} \sim S_{40}$	69.2	97.8	90.9	99.1	99.4	99.8
Pitch-scaling (-1%)	$S_{01} \sim S_{10}$	58.8	98.3	74.2	99.7	88.6	100
	$S_{11} \sim S_{20}$	62.6	97.2	78.7	98.4	91.2	99.4
	$S_{21} \sim S_{30}$	62.1	94.8	67.4	96.4	72.7	97.5
	$S_{31} \sim S_{40}$	62.8	95.7	79.8	97.8	93.2	98.7

TABLE VII
DETECTION RATES OF THE PROPOSED SCHEME (LAYER 1) AND THE METHOD IN [4], WHERE ODG $= -0.6$ FOR BOTH METHODS

Attacks	Host signals	DRs (%)	
		Method in [4]	Proposed
Noise	$S_{01} \sim S_{10}$	99.6	99.9
	$S_{11} \sim S_{20}$	97.3	100
	$S_{21} \sim S_{30}$	99.5	99.9
	$S_{31} \sim S_{40}$	99.7	99.7
HPF (1 kHz)	$S_{01} \sim S_{10}$	93.4	100
	$S_{11} \sim S_{20}$	88.1	100
	$S_{21} \sim S_{30}$	86.1	100
	$S_{31} \sim S_{40}$	87.4	100
Echo addition	$S_{01} \sim S_{10}$	99.4	100
	$S_{11} \sim S_{20}$	94.3	100
	$S_{21} \sim S_{30}$	99.7	100
	$S_{31} \sim S_{40}$	97.6	99.9
Re-sampling	$S_{01} \sim S_{10}$	100	100
	$S_{11} \sim S_{20}$	99.2	100
	$S_{21} \sim S_{30}$	100	100
	$S_{31} \sim S_{40}$	99.8	100
MP3 (64 kbps)	$S_{01} \sim S_{10}$	98.8	100
	$S_{11} \sim S_{20}$	98.1	100
	$S_{21} \sim S_{30}$	99.0	100
	$S_{31} \sim S_{40}$	98.7	99.6
AAC (64 kbps)	$S_{01} \sim S_{10}$	98.5	100
	$S_{11} \sim S_{20}$	99.2	100
	$S_{21} \sim S_{30}$	98.7	100
	$S_{31} \sim S_{40}$	99.1	99.5
Cropping	$S_{01} \sim S_{10}$	59.8	99.7
	$S_{11} \sim S_{20}$	55.2	99.9
	$S_{21} \sim S_{30}$	57.8	98.7
	$S_{31} \sim S_{40}$	56.1	99.4
Time-scaling (+1%)	$S_{01} \sim S_{10}$	42.1	99.9
	$S_{11} \sim S_{20}$	41.9	100
	$S_{21} \sim S_{30}$	52.7	99.5
	$S_{31} \sim S_{40}$	48.6	97.3
Time-scaling (-1%)	$S_{01} \sim S_{10}$	58.8	99.4
	$S_{11} \sim S_{20}$	58.1	99.0
	$S_{21} \sim S_{30}$	52.8	99.8
	$S_{31} \sim S_{40}$	55.2	100
Pitch-scaling (+1%)	$S_{01} \sim S_{10}$	52.9	97.7
	$S_{11} \sim S_{20}$	49.5	93.2
	$S_{21} \sim S_{30}$	50.1	94.1
	$S_{31} \sim S_{40}$	49.8	97.8
Pitch-scaling (-1%)	$S_{01} \sim S_{10}$	48.0	98.3
	$S_{11} \sim S_{20}$	45.7	97.2
	$S_{21} \sim S_{30}$	47.2	94.8
	$S_{31} \sim S_{40}$	48.8	95.7

embedding capacity of 10 bps and same perceptual quality of ODG $= -0.6$. Both methods achieve 100% DRs against closed-loop attack, amplitude attacks, low-pass filtering attacks, and compression attacks when bit rates are 128 kbps and 96 kbps. DRs against remaining attacks are listed in Table VII. From Table VII we can see that the method in [4] completely fails under de synchronization attacks. Compared to the method in [4], the proposed method achieves higher DRs against all the attacks.

It is noteworthy to mention here that the de-synchronization attacks such as cropping and time scaling cause misalignments in time domain. However, the effect of this misalignment is small in DCT domain. In addition to this, the computation of mean values at the decoding end further compensates the errors caused by the attacks. As a result, the proposed method achieves acceptable robustness against de-synchronization attacks.

V. CONCLUSION

In this paper, we proposed a novel multi-layer audio watermarking algorithm, which uses patchwork-based techniques for watermark embedding and decoding. The proposed algorithm can embed and extract watermark bits successfully from multiple layers. In the proposed algorithm, watermark bits are embedded into the audio signal by modifying its DCT coefficients. Embedding algorithm is designed in a way to ensure embedding watermarks in a particular layer does not affect the watermark bits embedded in the other layers. To increase the robustness of the proposed algorithm against attacks such as noise addition and re-quantization, an error buffer is introduced in the proposed algorithm. Furthermore, to withstand the attacks such as compression (MP3, AAC) attacks and filtering (high-pass, low-pass) attacks, watermark bits are only embedded into the frequency coefficients which are not severely affected by these attacks. In the proposed algorithm, to reduce the perceptual quality degradation caused by the addition of watermarks into multiple layers, a specifically designed DCT coefficient ordering process is included in the proposed algorithm. In addition to this, to improve the perceptual quality, a layer specific error buffer size is used in the embedding process. The simulation results demonstrate the robustness and perceptual quality of the proposed multi-layer watermarking algorithm.

REFERENCES

[1] G. Boato, F. G. B. Natale, and C. Fontanari, "Digital image tracing by sequential multiple watermarking," *IEEE Trans. Multimedia*, vol. 9, no. 4, pp. 677–686, Jun. 2007.
[2] G. Hua, J. Huang, Y. Q. Shi, and V. L. L. Thing, "Twenty years of digital audio watermarking—A comprehensive review," *Signal Process.*, vol. 128, pp. 222–242, Nov. 2016.
[3] W. N. Lie and L. C. Chang, "Robust and high-quality time-domain audio watermarking based on low-frequency amplitude modification," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 8, no. 1, pp. 46–59, Feb. 2006.
[4] N. K. Kalantari, M. A. Akhaee, S. M. Ahadi, and H. Amindavar, "Robust multiplicative patchwork method for audio watermarking," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 17, no. 6, pp. 1133–1141, Aug. 2009.
[5] H. Kang, K. Yamaguchi, B. Kurkoski, K. Yamaguchi, and K. Kobayashi, "Full-index-embedding patchwork algorithm for audio watermarking," *IEICE Trans. Inf. Syst.*, vol. E91-D, no. 11, pp. 2731–2734, Nov. 2008.
[6] Y. Xiang, I. Natgunanathan, S. Guo, W. Zhou, and S. Nahavandi, "Patchwork-based audio watermarking method robust to de-synchronization attacks," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 22, no. 9, pp. 1413–1423, Sep. 2014.
[7] I. Natgunanathan, Y. Xiang, Y. Rong, W. Zhou, and S. Guo, "Robust patchwork-based embedding and decoding scheme for digital audio watermarking," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 8, pp. 2232–2239, Oct. 2012.
[8] G. Hua, J. Goh, and V. L. L. Thing, "Time-spread echo-based audio watermarking with optimized imperceptibility and robustness," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 23, no. 2, pp. 227–239, Feb. 2015.
[9] G. Hua, J. Goh, and V. L. L. Thing, "Cepstral analysis for the application of echo-based audio watermark detection," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 9, pp. 1850–1861, Sep. 2015.
[10] H. Wang, R. Nishimura, Y. Suzuki, and L. Miao, "Fuzzy self-adaptive digital audio watermarking based on time-spread echo hiding," *Appl. Acoust.*, vol. 69, no. 10, pp. 868–874, Oct. 2008.

[11] O. T.-C. Chen and W.-C. Wu, "Highly robust, secure, and perceptual-quality echo hiding scheme," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 16, no. 3, pp. 629–638, Mar. 2008.
[12] B.-S. Ko, R. Nishimura, and Y. Suzuki, "Time-spread echo method for digital audio watermarking," *IEEE Trans. Multimedia*, vol. 7, no. 2, pp. 212–221, Apr. 2005.
[13] Y. Xiang, D. Peng, I. Natgunanathan, and W. Zhou, "Effective pseudonoise sequence and decoding function for imperceptibility and robustness enhancement in time-spread echo based audio watermarking," *IEEE Trans. Multimedia*, vol. 13, no. 1, pp. 2–13, Feb. 2011.
[14] Y. Xiang, I. Natgunanathan, D. Peng, W. Zhou, and S. Yu, "A dual-channel time-spread echo method for audio watermarking," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 2, pp. 383–392, Apr. 2012.
[15] H. J. Kim and Y. H. Choi, "A novel echo-hiding scheme with backward and forward kernels," *IEEE Trans. Circuit Syst. Video Technol.*, vol. 13, no. 8, pp. 885–889, Aug. 2003.
[16] Y. Xiang, I. Natgunanathan, Y. Rong, and S. Guo, "Spread spectrum based high embedding capacity watermarking method for audio signals," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 23, no. 12, pp. 2228–2237, Dec. 2015.
[17] I. J. Cox, J. Kilian, F. T. Leighton, and T. Shanon, "Secure spread spectrum watermarking for multimedia," *IEEE Trans. Image Process.*, vol. 6, no. 12, pp. 1673–1687, Dec. 1997.
[18] H. S. Malvar and D. A. Florencio, "Improved spread spectrum: A new modulation technique for robust watermarking," *IEEE Trans. Signal Process.*, vol. 51, no. 4, pp. 898–905, Apr. 2003.
[19] A. Valizadeh and Z. J. Wang, "Correlation-and-bit-aware spread spectrum embedding for data hiding," *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 2, pp. 267–282, Jun. 2011.
[20] P. Zhang, S. Xu, and H. Yang, "Robust audio watermarking based on extended improved spread spectrum with perceptual masking," *Int. J. Fuzzy Syst.*, vol. 14, no. 2, pp. 289–295, Jun. 2012.
[21] X. Zhang and Z. J. Wang, "Correlation-and-bit-aware multiplicative spread spectrum embedding for data hiding," in *Proc. 2013 IEEE Int. Workshop Inf. Forensics Security*, 2013, pp. 186–190.
[22] S. Kirbiz and B. Gunsel, "Robust audio watermark decoding by supervised learning," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2006, pp. 761–764.
[23] X. Wang, W. Qi, and P. Niu, "A new adaptive digital audio watermarking based on support vector regression," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 15, no. 8, pp. 2270–2277, Nov. 2007.
[24] D. Lakshmi, R. Ganesh, R. Marni, R. Prakash, and P. Arulmozhivarman, "SVM based effective watermarking scheme for embedding binary logo and audio signals in images," in *Proc. 2008 IEEE Region 10 Conf.*, 2008, pp. 1–5.
[25] C. M. Pun and X. C. Yuan, "Robust segments detector for desynchronization resilient audio watermarking," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 21, no. 11, pp. 2412–2424, Nov. 2013.
[26] K. Khaldi and A. O. Boudraa, "Audio watermarking via EMD," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 21, no. 3, pp. 675–680, Mar. 2013.
[27] A. Abrardo and M. Barni, "A new watermarking scheme based on antipodal binary dirty paper coding," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 9, pp. 1380–1393, Sep. 2014.
[28] S. C. Kushwaha, P. Das, and M. Chakraborty, "Multiple watermarking on digital audio based on DWT technique," in *Proc. Int. Conf. Commun. Signal Process.*, 2015, pp. 303–307.
[29] L. Tianchi, Y. Guangming, and W. Qi, "A multiple audio watermarking algorithm based on shear resisting DWT and LSB," in *Proc. Int. Conf. Netw. Comput.*, 2011, pp. 78–83.
[30] A. Ogihara, H. Murata, M. Iwata, and A. Shiozaki, "Multi-layer audio watermarking based on amplitude modification," in *Proc. 5th Int. Conf. Intell. Inf. Hiding Multimedia Signal Process.*, 2009, pp. 68–71.
[31] J. L. Wu and J. Shin, "Discrete cosine transform in error control coding," *IEEE Trans. Commun.*, vol. 43, no. 5, pp. 1857–1861, May 1995.
[32] ITU, *Methods for Objective Measurements of Perceived Audio Quality*, Rec. B.S. 1387, Int. Telecommun. Union, Geneva, Switzerland, 2001.
[33] K. Iwamura *et al.*, "Information hiding and its criteria for evaluation," *IEICE Trans. Inf. Syst.*, vol. E100-D, no. 1, pp. 2–12, Jan. 2017.
[34] C. Baras, N. Moreau, and P. Dymarski, "Controlling the inaudibility and maximizing the robustness in an audio annotation watermarking system," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 14, no. 5, pp. 1772–1782, Sep. 2006.

Iynkaran Natgunanathan received the B.Sc. Eng. (Hons.) degree in electronics and telecommunication engineering from the University of Moratuwa, Moratuwa, Sri Lanka, in 2007, and the Ph.D. degree from Deakin University, Burwood, VIC, Australia, in 2012. From 2006 to 2008, he was a Software Engineer with Millennium Information Technology (Pvt.), Ltd., Sri Lanka. He is currently a Research Fellow with the School of Information Technology, Deakin University. His research interests include digital watermarking, audio and image processing, telecommunication, and robotics.

Yong Xiang (SM'12) received the Ph.D. degree in electrical and electronic engineering from The University of Melbourne, Parkville, VIC, Australia. He is currently a Professor and the Director of the Artificial Intelligence and Data Analytics Research Cluster, School of Information Technology, Deakin University, Burwood, VIC, Australia. His research interests include information security and privacy, multimedia (speech/image/video) processing, wireless sensor networks and IoT, and biomedical signal processing. He has published 2 monographs, more than 90 refereed journal articles, and numerous conference papers in these areas. He is an Associate Editor of the IEEE SIGNAL PROCESSING LETTERS and the IEEE ACCESS. He was the Program Chair, the TPC Chair, the Symposium Chair, and the Session Chair for a number of international conferences.

Guang Hua received the B.Eng. degree in communication engineering from Wuhan University, Wuhan, China, in 2009, and the M.Sc. degree in signal processing and the Ph.D. degree in information engineering from Nanyang Technological University, Singapore, in 2010 and 2014, respectively. From July 2013 to November 2015, he was a Research Scientist in the Department of Cyber Security and Intelligence, Institute for Infocomm Research, Singapore. After that, he joined the School of Electrical and Electronic Engineering, Nanyang Technological University, as a Research Fellow until 2017. He is currently with the School of Electronic Information, Wuhan University, China. He has first authored more than 10 highly ranked IEEE journal and conference papers. He holds one Singapore patent. His research interests include general signal processing, applied convex optimization, digital filter design, and multimedia forensics and security.

Gleb Beliakov (M'08–SM'08) received the Ph.D. degree in physics and mathematics in Moscow, Russia, in 1992.

He worked at the Universities of Melbourne and South Australia, and is currently at Deakin University in Melbourne, Burwood, VIC, Australia. He is currently a Professor in the School of Information Technology, Deakin University. His research interests include the areas of aggregation operators, multivariate approximation, global optimization, and numerical computing. He is the author of nearly 200 research papers and two monographs in the mentioned areas, and a number of software packages. He was an Associate Editor of the IEEE TRANSACTIONS ON FUZZY SYSTEMS and *Fuzzy Sets and Systems* journals.

John Yearwood received the B.Sc. degree from Monash University, Clayton, VIC, Australia, the M.Sc. degree from Sydney University, Camperdown, NSW, Australia, and the Ph.D. degree from RMIT University, Melbourne, VIC, Australia. In 1989, he was with the School of Information Technology and Mathematical Science, University of Ballarat, Australia, as a Lecturer, where he was appointed as a Professor in 2007. He is currently a Professor and the Head of the School of Information Technology, Deakin University, Burwood, VIC, Australia. He has published more than 140 refereed journals, book chapters, and conference articles. His research interest includes modern optimization theory and techniques and their applications in pattern recognition, signal processing, and decision support systems.